

IMPLEMENTASI PEMBARUAN ALAMAT IP PADA DOMAIN NAME SYSTEM DINAMIS MENGGUNAKAN PYTHON

Yuda Nugrahadi¹, Sujoko Sumaryono², Addin Suwastomo³

¹Mahasiswa Jurusan Teknik Elektro dan Teknologi Informasi, FT UGM

²Dosen Jurusan Teknik Elektro dan Teknologi Informasi, FT UGM

Abstrak

DNS (Domain Name System) mempunyai peranan penting pada jaringan internet. Jaringan komputer atau internet pada dasarnya terdiri atas alamat-alamat IP, karena hal tersebut pengguna akan mengalami kesulitan dalam mengingat alamat-alamat tersebut. Dengan adanya DNS maka pengguna tidak perlu mengingat dalam bentuk alamat IP, cukup dengan nama domain saja, karena fungsi DNS adalah mentranslasikan nama domain menjadi alamat IP yang berupa digit angka dan sebaliknya. Di Indonesia, terdapat operator telekomunikasi yang memberikan layanan dengan fasilitas sebuah IP publik dinamis untuk pelanggannya. Dengan IP publik yang dinamis ini, pelanggan akan mengalami kesulitan jika ingin mempergunakannya untuk sebuah layanan. Hal ini karena setiap perubahan alamat IP publik maka record pada DNS juga harus ikut disesuaikan. Oleh karena itu dibutuhkan sebuah sistem yang bisa mentranslasikan domain ke alamat IP publik yang dinamis. Dengan cara ini pengguna tidak perlu mengubah alamat IP publik untuk domain yang digunakan setiap kali alamat IP publik tersebut berganti.

Kata kunci: DNS, dynamic DNS, python

1. Pendahuluan

Dengan mendapatkan sebuah IP publik maka pengguna dapat membuat sebuah layanan yang dapat diakses dari internet. Namun IP publik yang didapatkan oleh pengguna tersebut adalah IP publik yang bersifat dinamis, sehingga IP publik yang didapatkan tidak selalu tetap. Karena terus berubah maka akan menjadi sulit untuk mendaftarkan alamat IP tersebut ke DNS. Karena fungsinya adalah memetakan alamat IP ke nama domain maka setiap ada perubahan alamat IP diperlukan pembaruan alamat IP yang baru agar DNS dapat memetakan domain ke alamat IP yang baru.

Layanan *dynamic DNS* sudah dapat mengatasi masalah tersebut. Dengan menggunakan layanan *dynamic DNS* dimungkinkan untuk dilakukan pembaharuan alamat IP publik secara otomatis atas domain tertentu. Namun penyedia layanan *dynamic DNS* ini kebanyakan masih berada di luar negeri. Untuk pengguna dalam negeri, jika pemrograman, Python dikenal akan kesederhanaannya tetapi di balik itu, Python mendukung banyak pustaka yang tersimpan dalam modul-modul seperti pustaka untuk mendukung jaringan, antarmuka grafis, web (HTML, XML), akses basisdata (MySQL, PostgreSQL), dan berbagai hal lainnya [1]. Selain itu kode yang ditulis akan terlihat lebih

terjadi putusnya koneksi internet internasional maka secara total domain yang sudah didaftarkan di layanan *dynamic DNS* tersebut tidak dapat diakses. Layanan *dynamic DNS* terdiri atas DNS server dan updater. DNS server berfungsi untuk memetakan alamat IP publik ke domain dan sebaliknya sedangkan updater berfungsi untuk melakukan pengiriman alamat IP publik dinamis yang didapatkan pengguna ke DNS server.

Implementasi *updater dynamic DNS* ini dipilih dengan menggunakan bahasa pemrograman Python karena dapat berjalan pada berbagai macam sistem operasi (BSD, GNU/Linux, MacOS, Windows) khususnya yang berbasis unix tanpa harus mengubah kode sedikitpun dan tanpa menggunakan emulasi (virtualisasi). Hal inilah yang menjadi keunggulan Python. Selain itu kode Python mudah dibaca siapa saja, baik oleh pemula maupun oleh mereka yang sudah terbiasa dengan dunia pemrograman. Sebagai bahasa rapih karena Python menggunakan indentasi sebagai penentu sebuah blok kode.

Pada awalnya digunakan teknik *host table* yang menyimpan daftar seluruh nama komputer beserta alamat IP dan disimpan dalam berkas bernama HOSTS.TXT, kemudian setiap komputer mengambil berkas tersebut untuk digunakan [3]. Cara ini menjadi tidak efisien jika

terdapat 1000 host lebih untuk dicatat dan banyak host dalam jaringan yang memerlukan berkas tersebut untuk pengalamatan.

Paul Mockapertis mengusulkan sistem database yang terdistribusi yang dinamakan DNS (Domain Name System)[4][5]. Dengan model terdistribusi ini akan lebih mudah untuk melakukan pembaruan terhadap host yang terhubung ke internet. Selain untuk memetakan alamat IP dan hostname, DNS juga digunakan sebagai sarana bantu penyampaian e-mail.

HTTP mempunyai metode (atau sering disebut "verba") yang mengindikasikan tindakan yang harus dilaksanakan pada sumber daya yang teridentifikasi. Metode HTTP terdiri atas *CONNECT, OPTIONS, DELETE, GET, HEAD, POST, PUT, TRACE* [1].

2. Metode Pembaruan

Modem ADSL terlebih dahulu dikonfigurasi menjadi mode *bridge* sehingga yang melakukan *dial* bukanlah modem melainkan PC router.

PC router yang dipergunakan adalah sebuah PC dengan processor Pentium III, 128MB RAM, dan 2 buah kartu jaringan.

Aplikasi *updater* melakukan pengambilan alamat IP secara berkala yang kemudian mengirimkan alamat IP dinamis tersebut ke server DNS dinamis *daring.web.id*. Data yang dikirimkan bukan hanya alamat IP dinamis saja tetapi sekaligus dengan nama pengguna dan kata sandi sebagai autentikasi sebelum *record* DNS diubah. Pengiriman data tersebut dilakukan menggunakan metode HTTP GET.

3. Hasil dan pembahasan

Dalam menggunakan aplikasi *updater* ini modem ADSL harus dikonfigurasi menjadi mode *bridge* sehingga PC router yang melakukan *dial*. Dengan konfigurasi ini IP public akan terdapat pada PC router. Hal ini dilakukan karena aplikasi dijalankan pada PC router sehingga aplikasi bisa mendapatkan alamat IP publik dinamis yang diberikan penyedia jasa.

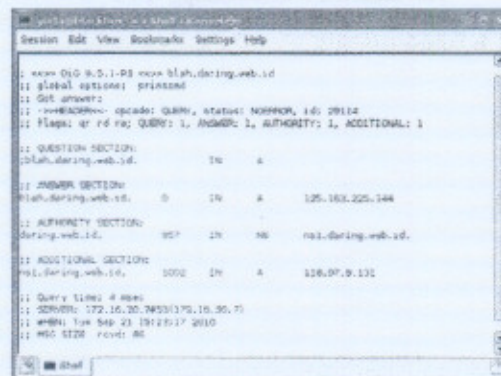
Aplikasi *updater* melakukan pengambilan alamat IP secara berkala

pada *interface* yang terhubung, dalam hal ini adalah *ppp0*. Setelah didapat alamat IP publik dinamis aplikasi akan melakukan pengiriman alamat IP tersebut ke layanan DNS dinamis *daring.web.id*.

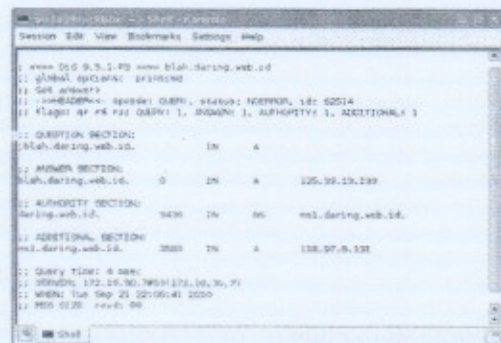
Pengiriman alamat IP dilakukan menggunakan metode HTTP GET karena metode tersebut menggunakan protokol HTTP yang umum digunakan sehingga tidak akan mengalami penutupan oleh *firewall*.

Untuk menguji proses pembaruan alamat IP publik dinamis sudah berjalan adalah dengan menggunakan DNS *query* menggunakan program seperti *dig* atau melalui situs yang menyediakan layanan *resolve* nama *domain* ke alamat IP seperti <http://domaintoip.com>. *Domain* yang digunakan untuk pengujian ini adalah *blah.daring.web.id*.

Pengujian dilakukan sebanyak 2 kali untuk memperlihatkan bahwa alamat IP publik yang digunakan untuk *domain* *blah.daring.web.id* memang berubah. Hasil pengujian pertama terlihat pada gambar 1 sedangkan hasil pengujian kedua terlihat pada gambar 2.



Gbr. 1 - Hasil Pengujian pertama



Gambar 2: Hasil Pengujian kedua

4. Kesimpulan

Aplikasi *updater* dapat melakukan pembaruan alamat IP dinamis atas *domain* atau *subdomain* yang dikehendaki.

Dengan kombinasi antara layanan DNS dinamis *daring.web.id* dan penggunaan aplikasi pembaruan ini maka alamat IP dinamis yang sebelumnya sulit untuk dimanfaatkan karena tingkat perubahannya sekarang dapat dimanfaatkan untuk layanan yang dapat diakses dari internet.

5. Referensi

- [1] Berners-Lee, Tim, & Fielding, R. 1999. RFC 2616: Hypertext Transfer Protocol – HTTP/1.1. <http://www.ietf.org/rfc/rfc2616.txt>. Diakses tanggal 27 Juni 2010, pukul 15.00 WIB.
- [2] Mockapetris, Paul. 1983. RFC 882: Domain Names – Concept and Facilities. <http://www.ietf.org/rfc/rfc882.txt>. Diakses tanggal 1 Mei 2010, pukul 10.15 WIB.
- [3] Kadir, Abdul. 2009. Dasar Pemrograman Python. Yogyakarta. Penerbit Andi
- [4] Liu, Cricket; & Albits, Paul. 2006. DNS and BIND Fifth Edition. USA. O'Reilly Media, Inc. [http://books.google.co.id/books?id=zkZN52WhG8sC&pg=PP1](http://books.google.co.id/books?id=zkZN52WhG8sC&pg=PP1&dq=DNS%20and%20bind&pg=PP1#v=onepage&q&f=false) &dq=DNS%20and%20bind&pg=PP1#v=onepage&q&f=false. Diakses tanggal 30 April 2010 pukul 10.00 WIB.
- [5] Mockapetris, Paul. 1983. RFC 882: Domain Names – Concept and Facilities. <http://www.ietf.org/rfc/rfc882.txt>. Diakses tanggal 1 Mei 2010, pukul 10.15 WIB.
- [6] Mockapetris, Paul. 1983. RFC 883: Domain Names – Implementation and Specification. <http://www.ietf.org/rfc/rfc883.txt>. Diakses tanggal 1 Mei 2010, pukul 10.17 WIB.
- [7] Purbo, Onno W. 2003. TCP/IP Standar, Desain, dan Implementasi. Jakarta. Penerbit PT. Elex Media Komputindo